UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/900,168	07/09/2001	Akira Kamiya	2001_0976A	5407

513 7590 03/14/2005

WENDEROTH, LIND & PONACK, L.L.P.
2033 K STREET N. W.
SUITE 800
WASHINGTON, DC 20006-1021

EXAMINER	
LEE, RICHARD J	

ART UNIT	PAPER NUMBER
2613	

DATE MAILED: 03/14/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/900,168	KAMIYA, AKIRA	
	Examiner	Art Unit	
	Richard Lee	2613	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 December 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1, 3-5, 7, 8* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-5,7 and 8* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *9/9/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. The request filed on December 3, 2004 for a Request for Continued Examination (RCE) is acceptable and a RCE has been established. An action on the RCE follows.

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 3, 4, 5, 7, and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kawakami of record (6,332,058) in view of Siong et al of record (6,028,632) and Haskell et al of record (5,159,447).

Kawakami discloses an information reproduction apparatus as shown in Figures 1 and 2, and the same multiple decoding method, in which a signal is composed of a plurality of encoded data is inputted, to simultaneously decode two or more of the data (see Figures 1 and 2), and multiple decoding apparatus receiving a signal composed of a plurality of encoded data for simultaneously decoding two or more of the data (see Figures 1 and 2) as claimed in claims 1 and 5, comprising substantially the same inputting the signal and extracting the two or more data to be decoded and reproduced (i.e., 18 of Figures 1 and 2); storing the extracted data in a buffer (i.e., 30 of Figure 1); distributing the data stored in the buffer (i.e., as provided by 40 of Figure 2 and see column 5, lines 46-54, column 7, lines 7-18) for each type (i.e., the MPEG stream of data as shown in Kawakami is based according to a specific type of video which includes inherent and specific header data, see column 5, lines 46-54, column 7, lines 7-18) and respectively storing the data in a plurality of separate buffers (i.e., 34 of Figure 2); controlling output of data stored in the plurality of separate buffers such that the data stored in the plurality of separate

buffers are associated with each other (i.e., as provided by 32 of Figure 2 and see column 5, lines 31-45); decoding, responsive to the controlling, the data stored in the plurality of separate buffers and outputting the decoded data (i.e., as provided by 22 of Figure 2, and see column 5, lines 31-45); reproduction controller (i.e., 24, 36 of Figure 2) for outputting various types of control information related to decoding and reproduction of the data; a data extractor (i.e., MPEG core server 18 of Figures 1 and 2) for receiving the signal for extracting the two or more data designated by the control information; a buffer (i.e., 20, 30 of Figure 2) storing the data extracted by the data extractor; a buffer manager (i.e., within core server 18 of Figures 1 and 2, and see column 5, lines 1-30) for controlling the buffer in accordance with the control information for the buffer; a data flow controller (i.e., 40 of Figure 2 and see column 5, lines 46-54, column 7, lines 7-18) for distributing the data stored in the buffer for each type and transferring the data in accordance with provided transfer conditions; a plurality of separate buffers (i.e., 34 of Figure 2) for respectively storing the data distributed and transferred by the data flow controller; a plurality of decoders (i.e., 22 of Figures 1 and 2) respectively corresponding to the plurality of separate buffers for decoding the data stored in the separate buffers and outputting the decoded data; and a decoding controller for selecting a separate buffer and a decoder (i.e., CPU group 36 outputs control signal 38 in response to a request from external controller 24, thereby selecting the desired information for decoding to the respective buffer and decoder, see column 5, lines 46-54, column 7, lines 7-38) which are used for the decoding, from among the plurality of separate buffers and the plurality of decoders in accordance with the control information, and outputting information related to the separate buffer selected by the decoding controller, the transfer conditions based on the separate buffer selected by the decoding controller, and an instruction to

start the decoding, respectively, to the separate buffer manager, the data flow controller, and the decoder selected by the decoding controller (i.e., controller 24 and CPU group 36 controls all the hardware structures, see columns 5-7).

Kawakami does not particularly disclose, though, the followings:

(a) a separate buffer manager for controlling the outputs of the plurality of separate buffers so as to be associated with each other in accordance with information for specifying the plurality of separate buffers as claimed in claim 1.

(b) the buffer manager outputs, when the buffer becomes full of the data, an overflow notification to the reproduction controller; the reproduction controller outputs, upon receipt of the overflow notification, an instruction to stop the data extraction to the data extractor, and outputs an initialization instruction to the decoding controller; the decoding controller outputs, upon receipt of the initialization instruction from the reproduction controller, an instruction to initialize all of the plurality of separate buffers to the separate buffer manager, outputs to the buffer manager an instruction to initialize the buffer, and respectively outputs instructions to stop the decoding to all of the plurality of decoders; the buffer manager initializes the buffer in accordance with the initialization instruction from the decoding controller; the separate buffer manager initializes all the plurality of separate buffers in accordance with the initialization instruction from the decoding controller; and all the processing which is stopped is resumed after all the buffer and the plurality of separate buffers are initialized as claimed in claim 1;

(c) the separate buffer manger outputs, when a specific separate buffer becomes full of the data, an overflow notification that the specific separate buffer overflows to the decoding controller, the decoding controller outputs, upon receipt of the overflow notification that the

separate buffer overflows, an instruction to stop the data transfer to the specific separate buffer to the data flow controller, an instruction to discard the data directed toward the specific separate buffer to the data flow controller, outputs an instruction to stop the decoding to a decoder corresponding to the specific separate buffer, and outputs to the separate buffer manager an instruction to initialize the specific separate buffer, the separate buffer manager initializes the specific separate buffer in accordance with the initialization instruction from the decoding controller, and all the processing which is stopped is resumed, and the discard of the data is released after the specific separate buffer is initialized as claimed in claims 3 and 4;

(d) when the buffer becomes full of the data, stopping extraction and decoding of the data, initializing all of the buffer and the plurality of separate buffers, and resuming all the processing which is stopped after all of the buffer and the plurality of separate buffers are initialized; when a specific separate buffer becomes full of the data, discarding the data directed toward the specific separate buffer, stopping the distribution of the data into the specific separate buffer and the decoding of the data stored in the specific separate buffer, initializing the specific separate buffer, and resuming all the processing which is stopped after the specific separate buffer is initialized, and releasing the discard of the data as claimed in claims 5, 7, and 8.

Regarding (a), it is noted that Kawakami does teach the particular use of a plurality of buffer managers (i.e., 32 of Figure 2) for controlling the outputs of each of the respective plurality of separate buffers 34, but and not particularly a separate buffer manager for controlling the outputs of the plurality of separate buffers as claimed. However, Siong et al discloses a multiple buffer and video decoder management system as shown in Figure 1, and teaches the general concept of the use of a separate buffer manager (i.e., 6 of Figure 1 and see column 3, line

56 to column 4, line 27) for controlling outputs of the plurality of separate buffers (i.e., 7-9 of Figure 1). Therefore, it would have been obvious to one of ordinary skill in the art, having the Kawakami and Siong et al references in front of him/her and the general knowledge of buffer management systems, would have had no difficulty in providing the separate buffer manager of Siong et al in place of the plurality of separate buffer managers 32 of Kawakami for the same well known single unit integrated processing and so that less hardware would be required for managing the buffers purposes as claimed.

Regarding (b) to (d), Haskell et al discloses a buffer control for variable bit rate channel as shown in Figures 1-4, and teaches the conventional notification of overflow situations associated with encoder and decoder buffers (see column 17, line 66 to column 18, line 13), and the particular termination of packets of data within the decoder as one way of preventing overflow in the buffers, thereby stopping decoding to the decoder, data extraction, data transfer to the specific buffer, and discarding data directed toward the specific buffer (see column 16, lines 27-39). It is noted that Haskell et al is however silent as to the initialization of the respective buffer components in response to the overflow notification and the subsequent resuming of the processing which was stopped after buffer initialization and the discard of the data is released after the buffer is initialized as claimed. But, it is considered obvious even without specific disclosure that once the packets are terminated within Haskell due to buffer overflow, the buffers of Haskell must be initialized since the existing data within the buffers are of no use and so that the buffers could be properly re-set. Further, after such buffer initialization and re-setting within Haskell, all processing will therefore be resumed, and the discarded data is released (i.e., the existing data in the buffer is of no use and therefore is released) after buffer

initialization. Therefore, it would have been obvious to one of ordinary skill in the art, having the Kawakami, Siong et al, and Haskell et al references in front of him/her and the general knowledge of video encoder and decoder buffer fullness, would have had no difficulty in providing the overflow notification, termination of packets of data within the decoder as one way of preventing overflow in the buffers, thereby stopping decoding to the decoder, data extraction, data transfer to the specific buffer, and discarding data directed toward the specific buffer as taught by Haskell as well as the obvious initialization of buffers upon receipt of an overflow notification and the subsequent resuming of the processing which was stopped after buffer initialization and the discard of the data is released after the buffer is initialized within Haskell for the multiple decoder of Kawakami so that the buffer manager, reproduction controller, decoding controller, and separate buffer manager of Kawakami may proper respond to the overflow notification for the same well known video decoder buffer overflow protection purposes as claimed.

4. Regarding the applicant's arguments at pages 6-8 of the amendment filed November 5, 2004 concerning in general that "... In amended claims 1 and 5, the above described process is carried out to allow a plurality of video or audio data contained in one MPEG transport stream to be concurrently decoded ... Kawakami sequentially reproduces information materials without causing a time lag, whereas the present invention simultaneously decodes a plurality of pieces of data. Paragraph 4 of the Office Action asserts that Kawakami discloses substantially the same multiple decoding apparatus as claimed in claim 1. This assertion is incorrect because Kawakami decodes data using a decoder in a one-to-one correspondence with the data ...In Kawakami, the time divisional multiplexing controller 40 sequentially reads divided and stored

information materials, and restores the read information materials to a piece of data so as to be decoded by one decoder. In contrast, as recited in amended claims 1 and 5, one MPEG transport stream is divided in to packets to be distributed to and decoded by a plurality of decoders ... Kawakami does not disclose that data is divided into portions which are transmitted to their corresponding decoders ...", the Examiner respectfully disagrees. The applicant's attention is directed to column 6, lines 42-59 of Kawakami for the particular teachings that compressed video and audio data are each divided into units called packets, and a packet PT (i.e. MPEG transport stream packet), is a minimum unit of components of an information material 14 when it is sent from a decoder buffer 34 to the associated decoder 22. Though the time divisional multiplexing controller 40 of Kawakami may sequentially read the information materials 14 from the DMA buffers 30, it is submitted that decoders 22 nevertheless simultaneously decodes the respective MPEG transport stream packets (i.e., packet PT) as provided by the respective decoder buffers 34. Kawakami teaches the desire to reproduce information materials 14 on many channels (i.e., simultaneously) by providing a parallel pipeline for the hard disk drives 20-1 to 20-5 (see column 5, lines 55-65), and since Kawakami also teaches a parallel pipeline processing of packet PTs within the respective decoders 22, it is hence clear that Kawakami provides substantially the same if not the same multiple decoding apparatus and method in which a signal composed of a plurality of encoded data is inputted, to simultaneously decode two or more of the data as claimed.

Regarding the applicant's arguments at page 9 of the amendment filed November 5, 2004 concerning in general that "... the data extractor 110 extracts data which coincides with set conditions from input data, and the extracted data is stored in the buffer 120 ... it is clear that

Siong fails to disclose or suggest a data extractor as recited in independent claim 5 ...", the Examiner wants to point out that the MPEG core server 18 of Kawakami nevertheless is considered substantially the same if not the same data extractor unit for receiving the signal for extracting the two or more data designated by the control information, as claimed. Specifically, Kawakami teaches that core server 8 receives the information materials 14 (i.e., two or more data) recorded on hard disk drives 20-1 to 20-5 and supplies the information materials 14 to a respective MPEG decoder in accordance with control signal 38 (see column 5, lines 10-23, lines 55-65). As such, it is submitted that the claimed features are rendered obvious in view of the combination of Kawakami, Siong et al, and Haskell et al.

Regarding the applicant's arguments at pages 9-11 of the amendment filed November 5, 2004 concerning in general that "... Haskell merely discloses, at col. 15, ll. 27-39, that a packet could be terminated in order to prevent decoder overflow. Thus, unlike amended claims 1 and 5, Haskell fails to disclose stopping decoding to the decoder, data extraction, and data transfer to the specific buffer, and discarding data directed toward the specific buffer ... However, Haskell et al does not teach any effect achieved by stopping the decoding of the decoder ... Haskell describes, at col. 16, ll. 27-39, that a packet could be terminated in order to prevent decoder overflow, but is silent as to the buffer initialization. Therefore, it is clear that Haskell fails to disclose or suggest the buffer initialization feature as recited in amended independent claims 1 and 5 ...", the Examiner wants to point out that once the packet is terminated within Haskell et al (see column 16, lines 27-39) due to overflow of the decoder buffer, this terminated packet is not decoded and as such decoding to the decoder is stopped, data extraction is stopped and data transfer to the specific buffer is stopped. And, it is submitted again that though Haskell et al is

silent as to the initialization of the respective buffer components in response to the overflow notification and the subsequent resuming of the processing which was stopped after buffer initialization and the discard of the data is released after the buffer is initialized as claimed, it is considered obvious even without specific disclosure that once the packets are terminated within Haskell due to buffer overflow, the buffers of Haskell must be initialized since the existing data within the buffers are of no use and so that the buffers could be properly re-set. Further, after such buffer initialization and re-setting within Haskell, all processing will therefore be resumed, and the discarded data is released (i.e., the existing data in the buffer is of no use and therefore is released) after buffer initialization. For the above reasons, it is further submitted that the combination of Kawakami, Siong et al, and Haskell et al renders obvious the claimed invention.

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Richard Lee whose telephone number is (571) 272-7333. The Examiner can normally be reached on Monday to Friday from 8:00 a.m. to 5:30 p.m, with alternate Fridays off.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group customer service whose telephone number is (703) 306-0377.

RICHARD LEE
PRIMARY EXAMINER

Richard Lee/rl

3/4/05